apl. Prof. Dr. Thomas Noll
Philipp Berger, Matthias Volk

Compiler Construction 2018/19

— Exercise Sheet 10 —

Hand in until January 7th, 2019 before the exercise class.

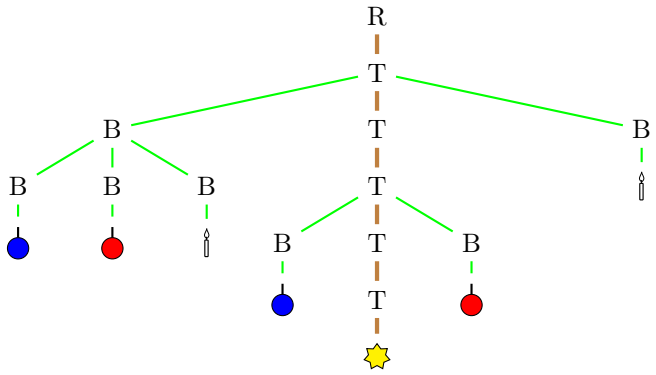
Exercise 1 (6 Points)

In the following we consider a grammar G whose derivation tree visualizes a christmas tree.

Let $G = (N, \Sigma, P, R)$ be the CFG with $N = \{R, T, B\}$ and $\Sigma = \{$  $,$  $,$ $,$  $\}$ [1]. The productions P are as follows:

$$R \to T$$
$$T \to BTB \mid T \mid$$ 
$$B \to BBB \mid BB \mid B \mid$$ $\mid$  $\mid$

An example christmas tree (upside down with the root at the top) would look as follows:



Give an attribute grammar $\mathfrak{A} = (G, E, V)$ for G such that the obtained christmas trees (i.e., derivation trees) satisfy the following restrictions:

(i) The distance from  to the start symbol R is greater than the distance between every other leaf and R.

(ii) The number of on the left and right hand side of the tree (split by the trunk T) are equal.

(iii) When traversing the tree in a pre-order manner the  and  are encountered alternatingly and the first visited ornament is  .
Hint: pre-order is a depth-first traversal where first the node, then the left subtree and last the right subtree are visited.

As before, you may use a synthesized Boolean attribute b such that: A word w is in the language of $\mathfrak{A} = (G, E, V)$ iff $w \in L(G)$ and, at the root of a derivation tree, $b.0 = \text{true}$.

The given example christmas tree satisfies all three restrictions as can be easily seen:

(i) The distance from  to the start symbol R is 6 whereas the largest other distance is 5.

(ii) On the left hand side there is one  and on the right hand side there is one  as well.

(iii) The traversal yields the following (correct) order of ornaments: ,  ,  , .

[1] For ease of notation you can also use $\Sigma = \{star, candle, red, blue\}$.

Exercise 2 (4 Points)

Consider the following intermediate code:

```
    ⋮
2:  LIT(-1);
3:  LOAD(1, 3);      (dif, off)
4:  LT;
5:  JFALSE(7);
6:  CALL(6, 1, 3);   (ca, dif, loc)
7:  RET;
```

Give the next four states of the abstract machine starting in:

$$(ca, d, p) := (3, 7 : -1, 6 : 5 : 42 : 6 : 6 : 6 : 7 : 6 : 30 : 1 : 2 : -3 : 4 : 9 : 3 : \ldots)$$

Recall that the procedure stack has the form:

sl	dl	ra	v_1	$\ldots$	v_n	$\ldots$

and the *base*-function is defined as:

$$\begin{aligned} base(p, 0) &:= 1 \\ base(p, dif + 1) &:= base(p, dif) + p.base(p, dif) \end{aligned}$$

Exercise 3 (4 Points)

In addition to `while`-loops we want to have `for`-loops with implicit declaration of the counter variable in our example programming language:

$$\texttt{for (var } X := A \texttt{ ; } B \texttt{ ; } C_1 \texttt{) } C_2$$

(a) Extend the translation function ct accordingly. You may assume that the variable X is already declared, i.e., it is $update(var\ X, st, l)$ with st the symbol table and l the current level.

(b) Generate intermediate code for

```
for (var x := 0; x < 10; x := x + 1) P()
```

without parameters for the `CALL` instruction generated for `P()`.

We wish you a merry Christmas and a Happy New Year!